N69-14024

CASE FILE COPY

NASA TT F-12,000

SELECTION OF AN OPTIMUM MEANS OF REALIZING CONTROL ALGORITHMS FOR AIRCRAFT COMPUTERS

N.T. Berezyuk

Translation of: 'Vybor Optimal'nogo Puti Realizatsii Algoritmov Upravlyayushchimi Mashinami Letatel'nykh Apparatov
In: Samoletostroyeniye i Tekhnika Vozdushnogo Flota (Aircraft Construction and Air-Fleet Equipment), No. II. Edited by Yu.N. Alekseyev. Kharkov, Kharkov State University Press, 1967, pp. 45-48.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546 DECEMBER 1968

SELECTION OF AN OPTIMUM MEANS OF REALIZING CONTROL ALGORITHMS FOR AIRCRAFT COMPUTERS

N.T. Berezyuk

ABSTRACT: The article develops a method for determining an algorithm involving the least computational labor in the realization of the transfer functions of automatic control systems on control computers. The method is based on the theory of graphs, and the results are given in the form of theorems.

Let us examine the following algorithm: /45*

$$y(n) = a_0\psi(n) + a_1\psi(n-1) + \ldots + a_k\psi(n-k) - b_1y(n-1) - b_2y(n-2) - \ldots - b_ky(n-k). \quad (1)$$

Algorithm (1) is often encountered when there is need for realizing transfer functions of certain automatic control systems on control computers. Calling $b_1 = a_{k+1}$, $b_2 = a_{k+2}, \ldots, b_k = a_{2k}$ in (1), we have the following:

$$y(n) = a_0\psi(n) + a_1\psi(n-1) + \ldots + a_k\psi(n-k) - a_{k+1}y(n-1) - \ldots - a_{2k}y(n-k).$$

Let us formulate the following lemma.

Lemma I. In any algorithm of the type in (1), the following dependences can be obtained:

$$\begin{aligned} a_1 &= r_0a_0, \\ a_2 &= r_1a_1, \\ &\ldots\ldots\ldots \\ a_{2k} &= r_{2k-1}a_{2k-1}, \end{aligned}$$

if the coefficients $a_0, a_1, \ldots, a_{2k}$ are real numbers.

---

* Numbers in the margin indicate pagination in the foreign text.

Proof. If the coefficients $a_0, a_1, \ldots, a_{2k}$ are real numbers, then we can always find the ratio between two real numbers, which will also be a real number, i.e.,

$$r_0 = \frac{a_1}{a_0},$$
$$r_1 = \frac{a_2}{a_1},$$
$$\ldots\ldots\ldots\ldots$$
$$r_{2k-1} = \frac{a_{2k}}{a_{2k-1}},$$

which proves the lemma.

Considering the lemma presented, (1) can be rewritten in the following way:

$$y(n) = [\ldots[[-y(n-k)\cdot r_{2k-1} - y(n-k-1)]\, r_{2k-2} - \ldots - y(n-1)]\, r_k - \psi(n-k)]\, r_{k-1} + \psi(n-k-1)]\, r_{k-2} + \ldots + \psi(n-1)]\, r_0 + \psi(n)]\, a_0 \qquad (2)$$

The number of operations in (1) and (2) is the same. However, less machine time is spent in realizing (2), since no time is spent here in transferring intermediary results. We will show this in the form of a schematic diagram, having first considered certain definitions. /46

Definition I. We will call the bending graph of the algorithm a tree[1] whose roots are the final results and whose vertices are the initial data, if the amount of final results is less than the

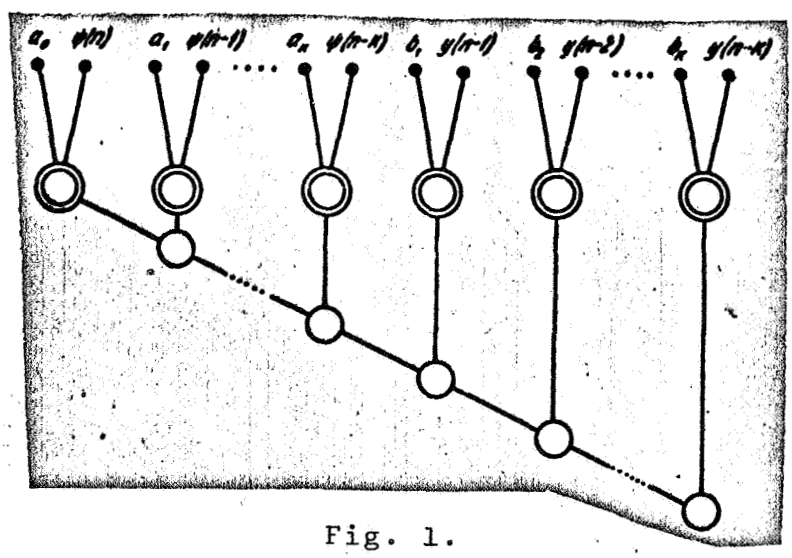

Fig. 1.

[1] Berzh, K.: Teoriya grafov i yeye primeneniye. (Graph Theory and its Applications). Foreign Language Publishing House, 1962.

amount of initial data. Moreover, we will designate the addition (substraction) operation in the schematic diagram by a single circle, multiplication by two circles, and an operation without name by one circle intersected by a line, while the initial data will be designated by large dots.

Definition II. If the number of final results in the algorithm is greater than the amount of original data, we will call the unfolding graph for such an algorithm a tree whose roots are the initial data and whose vertices are the final results.

The following obvious lemma follows from these definitions.

Lemma II. Any algorithm which has more final results than initial data can be represented by an unfolding graph, and any algorithm whose number of final results is less than the amount of initial data can be represented by a bending graph.

All the algorithms of the type in (1) can be represented by bending graphs, and the root of the tree is one final result.

Let us map out the bending graphs for algorithms (1) and (2) (see Figures 1 and 2). Having analyzed them, we can see that the bending graph for (2) bends toward the root in a straight line, while the graph in Figure 1 does not have any straight-line path between the left upper vertex and the root. However, any deviation from a straight bending line means conputation of intermediary results which must be preserved in the storage of the computer, and a certain amount of time must be spent in transferring these results. Subsequently, (2) is realized more quickly than is (1) by the control computer.

Let us formulate the following theorem. /4

Theorem. In any algorithm which is represented by a bending or unfolding graph, there is an optimal path (from the point of view

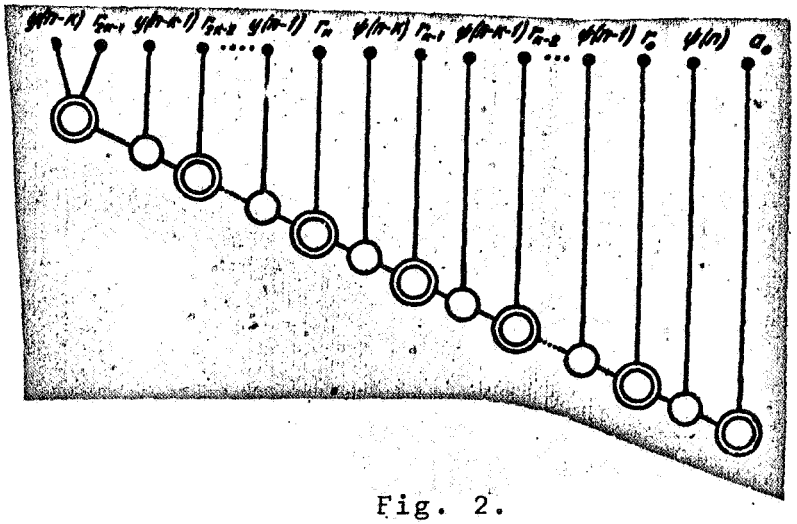

Fig. 2.

of best realization of the control computer) from the vertices of a tree to its root, if it is a bending graph, and from the root to the vertices, if it is an unfolding graph.

Proof. Let there be several operations A, B, C, ..., through which we should pass from the vertices to the root of a tree (we will prove the theorem for the bending case). For each pair of operations (A, B), we know the labor involved in its realization (by labor we mean the number of addition operations in Operations A and B plus the number of intermediary computation results which must be preserved in storage during the realization of A and B).

The graph of the least labor-involvement should be a tree, since, if it contained a cycle, we should be able to isolate one of the units of this cycle so that the operations still remain united. Consequently, we must draw m-1 lines in order to unite m operations.

We will show that the optimal path can be drawn by using a simple rule for the labor involved in realizing the operations. First of all, we will connect two operations with the least labor-involving connecting link (we will consider that the labor in the operations occurs in a line which connects them). At each of the following steps, we will add the least amount of labor in the links ω_i; when this is connected to the already-constructed branches, no cycle is developed. If there are several links involving the same amount of labor, we will select any one of them. Each tree O which is thus constructed will be called an optimal one. The amount of labor involved in it is equal to the sum of the amounts of labor involved in the separate links:

$$l(O) = l(\omega_1) + l(\omega_2) + \ldots + l(\omega_{m-1}).$$

Let us prove that there is no other tree L connecting the same vertices (the root with the extreme left vertex) which can involve a lesser amount of labor than the optimal tree O. Let L be the least labor-involving tree connecting the root to the vertices, and let O be any optimal tree. We will assume that the branches ω_1, ω_2,... of the optimal tree O are numbered in the same order in which they were connected in the construction of O. If the least labor-involving tree L does not coincide with O, then O has at least one /48
branch which does not belong to L. Let $\omega_i = (A, B)$ be the first such branch, and let $\Phi(A, B)$ be the circuit of graph L which connects the vertices of A and B. If the branch ω_i is added to L, then the graph of $L + \omega_i$ will contain the cycle $C = \omega_i + \Phi(A, B)$, and, since O does not have any cycles, cycle C should contain at least one branch which does not belong to O. Having removed this branch ω_i', we can obtain the following tree:

$$L' = L + \omega_i - \omega_i'.$$

This tree has the same number of vertices as does L, and the amount of labor involved in it is equal to

$$t(L') = t(L) + t(\omega_i) - t(\omega_i)$$

Since L involves the least possible amount of labor, we find that

$$t(\omega_i) \geqslant t(\omega_i').$$

However, ω_i was the least labor-involving link when it was added to $\omega_1, \omega_2, \ldots, \omega_{i-1}$ so that no cycles were obtained. Since we also do not obtain any cycle in adding the branch ω_i' to these branches, we can see that

$$t(\omega_i) = t(\omega_i'),$$

Subsequently, L' also involves the minimum amount of labor, i.e.,

$$t(L) = t(L').$$

Thus, we have found another tree L' involving the minimum amount of labor, which has one more branch than does L. We can repeat this operation until we finally obtain the connecting tree involving the minimum amount of labor and coinciding with O. Thus, O, as well as all the other optimal trees, actually involve the minimum amount of labor, which proves our theorem.

Translated for the National Aeronautics and Space Administration by:
Aztec School of Languages, Inc.,
Research Translation Division (74)
Acton, Massachusetts.
NASw-1692.